PanoInserts: Mobile Spatial Teleconferencing

Fabrizio Pece, William Steptoe, Fabian Wanner, Simon Julier,
Tim Weyrich, Jan Kautz, Anthony Steed
Department of Computer Science, University College London, UK
$\{f.pece, w.steptoe, s.julier, f.wanner, t.weyrich, j.kautz, a.steed\}$ @ $ucl.ac.uk$

Figure 1. A typical PanoInserts session. Two cameras, pointing at two users, are tracked using image features. Another two cameras, pointing at a white wall and a white-board, are tracked more crudely using a marker-based method.

ABSTRACT
We present PanoInserts: a novel teleconferencing system that uses smartphone cameras to create a surround representation of meeting places. We take a static panoramic image of a location into which we insert live videos from smartphones. We use a combination of marker- and image-based tracking to position the video inserts within the panorama, and transmit this representation to a remote viewer. We conduct a user study comparing our system with fully-panoramic video and conventional webcam video conferencing for two spatial reasoning tasks. Results indicate that our system performs comparably with fully-panoramic video, and better than webcam video conferencing in tasks that require an accurate surrounding representation of the remote space. We discuss the representational properties and usability of varying video presentations, exploring how they are perceived and how they influence users when performing spatial reasoning tasks.

Author Keywords
Mixed reality; teleconferencing; telepresence; remote collaboration; mobile phones; panoramas; camera tracking.

ACM Classification Keywords
H.4.3 Communications Applications: Computer conferencing, teleconferencing, and videoconferencing

General Terms
Experimentation, Human Factors, Performance.

INTRODUCTION
The quality and pervasiveness of cameras on mobile devices continues to increase. Most new laptops have a built-in camera, and most new smartphones and tablet-style devices have both front- and rear-mounted cameras. Rear-mounted cameras on mobile devices aim to replace or supplement the use of a point-and-shoot camera, while front-mounted and laptop cameras are often used for face-to-face video conferencing.

Mobile devices have enabled portable video teleconferencing. Due to the portable nature of the devices, users may move around their environment and reposition cameras freely. In contrast, highly-developed video conferencing systems such as Cisco TelePresence are designed to support group collaboration, and feature multiple cameras and displays to achieve gaze awareness and a sense of space. However, such systems require equipment to be installed in a dedicated meeting room and also impose constraints on where participants position themselves to maintain gaze awareness during communication [7]. Panoramic video conferencing as presented in [21] uses omnidirectional cameras such as the PointGrey Research LadyBug3, which capture a surrounding representation of a remote space and the people within.

The high-end systems described above are both expensive and lack portability, while the ubiquitous webcam-style video chat cannot easily transmit spatial relationships between several people or objects due to cameras typically having narrow fields of view. This paper introduces a system that we call "PanoInserts", which aims to support portable spatial video conferencing that lies between these two approaches in terms of both spatiality and accessibility. We aim to support meetings and other small-group interactions using only common personal devices communicating over the Internet. The system captures and transmits the visual representation of a real-world location and the people within for display to a remote viewer. It takes advantage of the pervasiveness of smartphones to create hybrid surround video communication in which a static panorama is augmented with live video inserts. As our system uses readily-available personal mobile devices, it can be rapidly configured and initiated, and lends itself to ad-hoc and spontaneous telecollaboration scenarios.

To outline the system's typical usage, imagine a typical video-conferencing session in which a group of people (the "locals") in one city would like to have a technical discussion with a colleague located in another city (the "visitor"). In the minutes prior to the conferencing session, one of the locals captures a panorama of the meeting room using built-in software on their smartphone. Subsequently, each local places their own smartphone in front of them so that its front camera points towards their seated position and the rear camera points at a marker (see Figure 4(a)). The visitor receives the live video streams from all locals' smartphones registered on the captured panorama. The visitor receives a surrounding representation of the meeting space, and hence can see the locals' seating arrangement and where each person is looking. During the discussion, the visitor asks the locals to draw a diagram to clarify some technical details. One of the locals repositions her phone to point at a white-board located in the meeting room and walks over to draw the diagram. The video-feed from the moving smartphone camera is tracked and re-registered within the panorama to present a live view of the white-board. Meanwhile, one of the still-seated local explains the diagram. The visitor can see both points of interest in the transmitted panoramic representation of the room.

The remainder of the paper is structured as follows. We cover related work including spatiality in video-mediated communication, panorama construction, and image alignment. We then detail the technical implementation of our system, including camera tracking, image registration, and rendering. Our novel approach makes use of commonly-available devices to achieve surrounding video conferencing for small-group interaction. We then present a user study addressing the fundamental implications for spatial perception over three video display modes: webcam, fully-panoramic, and our system. We show that PanoInserts provides a good compromise in terms of both spatiality and accessibility between expensive fully-panoramic video and conventional webcam conferencing. Finally, we discuss implications and design considerations for varying spatial forms of video conferencing, exploring how they are perceived and how they influence users when performing spatial reasoning tasks.

BACKGROUND

Spatiality in Video Conferencing

Spatiality in mediated communication is the degree to which a system supports fundamental properties such as movement, distance, containment, topology and a shared frame of reference such as a Cartesian coordinate system [3]. A telecommunications medium supporting a high-degree of spatiality, for example collaborative immersive virtual environments, presents a shared space in which all users observe, from their perspective, the same extents, relative positions, and orientations [2]. Practically, this implies that spatial cues such as gesture and gaze can be both performed and observed similarly to as they can be in reality. In contrast, webcam video conferencing presents portions of physical space that can constrain these spatial cues thereby hindering spatial perception and limiting gaze awareness [12]. High-end video-based telepresence systems such as [13] are able to support spatial cues and gaze awareness, but they require specific technology to be installed in dedicated rooms, reducing their potential for more ad-hoc or location-specific teleconferencing. There have also been several novel mixed-reality approaches including [4, 29, 24] that have demonstrated the importance of supporting spatial cues in telecommunication.

Panoramas offer a mode of video-mediated communication that can potentially foster a high-degree of spatial awareness. A surrounding representation of a remote space and the people within such as presented in [10] may overcome limitations associated with narrow field-of-view cameras. Panoramic cameras, also referred to as omnidirectional, such as the PointGrey Research Ladybug3 provide high-quality images with good sampling over the full panorama. Such cameras typically assume simple cylinder, sphere or cube proxy geometry for the scene, onto which all video is projected. Alternatives, providing lower and more uneven spatial resolution, are catadioptric systems or wide angle 'fish-eye' lenses and a single camera. Commercial systems for teleconferencing using such lenses include the Polycom CX5000. To augment the relatively low panoramic resolution, these systems can be augmented with scenario-specific video inserts [8].

As a basis for video-mediated communication, panoramas have not been thoroughly investigated from an HCI perspective. Mulloni et al. [19] explored the influence of varying panoramic representation on how users are able to locate objects in the image. Users achieved higher task performance using a simple frontal rectangular representation than a faithful (spherical) representation when performing this specific task. Our user study assesses the fundamental benefit and usability of panoramic display for telecollaboration by comparing three classes of capture featuring varying degrees of spatiality.

Construction of Panoramas

Panoramas are an attractive basis for videoconferencing as they provide a full 360° view of an environment in a single image. There are two main classes of methods to construct a panorama. The first class is based on special hardware, and includes solutions based on well-calibrated cameras [10] or special camera and mirror arrangements [17]. The second class is based on image-based algorithms, and includes registration of multiple videos [1, 23] or stitching of overlapping still images [26, 28]. While the first class of methods provides a fast and reliable solution to construct panoramas, its accessibility is limited by the high costs of the hardware. Image-based methods offer an accessible solution to construct panoramas that can be easily employed on a vast range of devices, including mobile phones. For this reason we decided to employ an image-based algorithm for constructing the static panoramas to be used in PanoInserts.

Image-based construction of panoramic imagery generally follows a two-step process. First, the arrangement of images to cover the panorama is discovered. Finding the arrangement of images is usually pairwise solved, by either direct or feature-based methods. Direct methods, such as the one proposed by Suen *et al.* [25], search over the space of possible transformations between image coordinates to find the

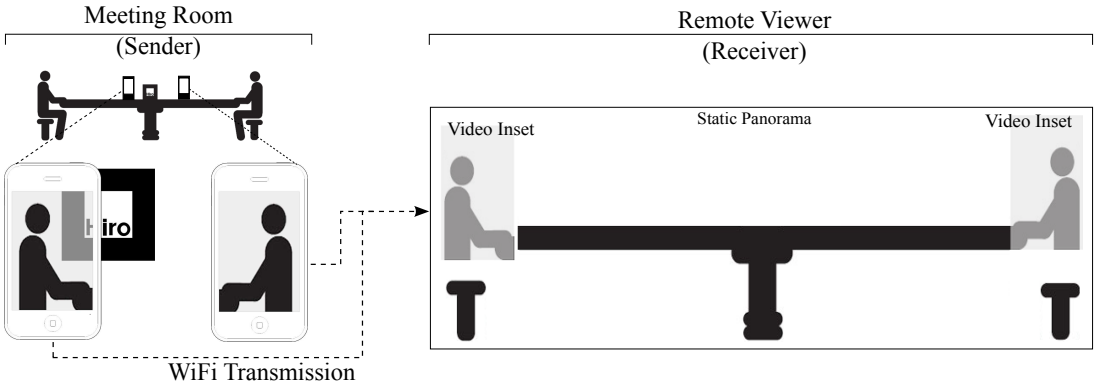


Figure 2. Architecture overview. In the meeting room, the smartphone on the left is performing marker-based camera tracking and transmission of both camera pose and video, while the smartphone on the right is streaming only video. The remote viewer, which runs on a standard PC, receives this information and a) inserts a video stream based on the rough marker-based location (on the left) and b) performs feature-based camera tracking and accurately positions the corresponding video (on the right). Both videos are overlaid onto the previously captured static panorama of the meeting room.

Figure 3. Static cube-map panorama. Note the absence of furniture.

minimum pixel-to-pixel dissimilarities between the two images. Feature-based methods [6] use a sparse set of features to find correspondences between two images, from which they compute a transformation of image coordinates between the two views. Subsequently, images are combined to recover the final mosaic. The combination phase may include correcting for variations in lighting, color balance, and exposure. These techniques are readily available on smartphones. Diverdi *et al.* presented the Envisor system [9] to construct a cube-map panorama by tracking SURF features, and Wagner *et al.* presented a system for constructing cylindrical panoramas by tracking FAST features [30].

Image Alignment

PanoInserts dynamically aligns video streams within a static panorama. Such image alignment can be achieved through a range of techniques including direct methods and feature correspondences, see [27] for a review. Direct methods search over the space of possible transformations between image coordinates to find the minimum pixel-to-pixel dissimilarities between the two images. Feature-based methods use a sparse set of image features locations to find correspondences between the two images and then compute a transformation of image coordinates between the two. While the former is computationally expensive, the latter can be often performed at interactive rates with comparable results.

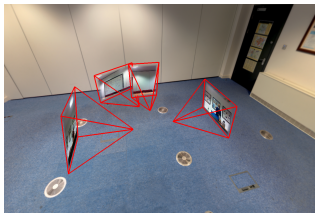
Detection of scale- and viewpoint-independent image features is a powerful tool to match information across images in order to find correspondences [22, 18]. In general, for a pair of images taken from partially-overlapping viewpoints, affine-invariant feature descriptors such as the Scale-Invariant Feature Transform (SIFT) descriptor [16] can be used to estimate a transformation that maps one view to the other. SIFT is effective for image registration and stitching [15, 27]. However, feature correspondence is usually an ambiguous, error-prone task, and therefore robust statistical techniques such as the least median of squares (LMedS) algorithm [20] or the random sample consensus (RANSAC) refinement [11] are used to reject erroneous matches across images and to reduce the probability of estimating erroneous transformations. We use SIFT and RANSAC in our system implementation to align the live video streams to a static panorama.

ARCHITECTURE OVERVIEW

Our system design is motivated by the goals of accessibility and practicality. The system should be accessible in the sense that a meeting place should not require cumbersome tracking equipment, cameras, or dedicated networks. Rather, the required hardware should be commonly available smartphones and computers connected to the Internet. The system should be practical, meaning that it should be configurable in less than five minutes and should be dynamically reconfigurable during use. This implies that users are able to connect, disconnect and reposition smartphones during the session. Allowing repositioning is particularly useful in situations where people are moving around the environment or when there are fewer available cameras than there are potential points of interest.

We use the video acquired from mobile phone cameras to transmit and dynamically insert views of the remote location within a static panorama. Our system comprises of three main modules: camera tracking, transmission and display. Figure 2 illustrates an overview of the system. The sender side features gross camera tracking based on marker (phone on the left in the figure) and transmission of both camera poses and video

(a) Configuration of four smartphone cameras around a marker.



(b) 3D positions of the cameras estimated from marker tracking.

Figure 4. PanoInserts marker-based tracking.

(a) Marker-based tracking.

(b) Feature-based tracking.

Figure 5. Results from different camera tracking methods.

streams. The receiver side is responsible for computing an accurate feature-based camera tracking and receiving, integrating and displaying together multiple videos from multiple cameras. In addition to this, our system requires a preliminary stage for acquisition of panoramas. This additional step can be performed by using any desired software, including additional software running directly on the phone.

The software running on the smartphones (i.e. the sending side) was written using ARToolkit for iOS and runs on devices running iOS4 or higher. The receiver-side software runs on PCs running Windows XP or higher, and uses the OpenFrameworks framework , which uses OpenGL for rendering. Finally, for the feature-based camera tracking we employed OpenCV and the SiftGPU package [31], a GPU implementation of the SIFT algorithm.

Construction of Panoramas

Many tools exist to assist in the construction of panoramas. While PanoInserts does not constrain the construction to any specific technique, it assumes that the panorama is available as a cube map, for display purposes. This, however, is not a limitation of the system, as conversion between panorama types can be easily performed. For the example scene shown in this paper, as well as for the user study we run with the system, we used a cube-map with six faces each 2048×2048 in resolution (see Figure 3). The panorama was assembled from 36 images captured with a Nikon D200 camera. These were stitched together using the PTGui software and exported as a cube-map. However, the panorama could have been built also with software readily available on the phone, such as Microsoft's Photosynth .

Camera Tracking

The system relies on two tracking approaches to ensure that the camera frame is displayed correctly within the panorama. The system's preferred choice of tracking is a feature-based tracker that is run on the receiver. This approach is used when enough image features can be extracted from the video streams. The other approach is based on a single marker, and it is used during the system setup or when the more accurate feature-based tracker fails (e.g., featureless areas or poor video quality). Our system supports both automatic and manual selection of the tracking type. Users can either manually switch between tracking techniques by touching the screen, or have the system automatically choose the best tracking solution. If automatic selection is enabled, the system uses the device accelerometer to assess whether the unit is moving or not, tracking the marker only when the phone is static.

Marker-based Tracking

Ideally, we would like to track the cameras solely by registering the images captured against the panorama, as this would allow the users in the environment to have full control over the cameras. However, there are several barriers in doing this. First of all, our panoramas are only roughly accurate: furniture and other objects might move or the lighting might change. Second, our envisaged capture spaces (i.e. indoor scenes) often contain large feature-less areas (e.g. white walls in Figure 1) which would not be amenable to direct or feature-based image alignment methods. Third, our scenes contain moving humans and other objects that move and change appearance (e.g the white board, which is on wheels, and the locals in Figure 1). In addition to this, we note that the quality of video available on mobile phones is usually low: under motion, the image is blurred and focusing and exposure balancing are slow.

Whilst some of these issues could be tackled by integrating other forms of camera tracking such as built in accelerometer and gyroscopic data, this is not a robust option over long periods. Such solutions tend to accumulate large tracking error over time. Instead, we decided to employ a marker based camera tracking that computes a gross camera pose estimation. Such estimation is enough to initially display the video frames in their correct location, with a relatively small error, and can be computed at interactive rates (Figure 5(a)). We exploit the fact that recent phones, such as the iPhone 4, have two cameras. This allows us to stream the video to augment the panorama from the front (display-side) camera, and to track the marker using the rear-side camera. We decided to employ the front camera video for the streaming so that the users can see the video that is being transmitted. Our system only requires a single marker in the environment, placed roughly in the center of the remote location (Figure 4). It is important to note that placing the marker roughly in the center of the remote location ensures that all the cameras that can see the marker roughly share an optical center. If the marker is also at the center of the panorama, then this guarantees that all the cameras will fit to the panorama.

Feature-based Tracking

Video registration based solely on marker-based tracking is only roughly accurate, resulting in a crude camera pose estimation. The next stage, then, is to refine such estimation by employing a more precise feature-based tracking algorithm

(Figure 5(b)). This step effectively means registering the camera image to the relevant face(s) of the cube-map. The registration requires the estimation of a homography that maps the video frame into the face of the cube-map that has most overlap. To find this homography, we robustly estimate the features matching within two views employing SIFT features and RANSAC refinement. We opted for SIFT descriptors as they are independent to different geometric transformations (scaling, rotation and translation), they are invariant to uniform scaling and orientation, and they also provide a very robust match across a large range of additional of noise and change in illumination.

When setting up the system, we pre-calculate and store SIFT descriptors for each of the six cube-map faces. As a new video image is received, from the last rough camera position given by the marker tracking we can filter out some of these SIFT descriptors from consideration to help removing false matches due to room symmetry and repeating elements. We then extract the features from the received frame and calculate the number of matches of these features against the filtered sets for all six cube-map faces. We take the face with the largest number of matches and refine the corresponding matches using the RANSAC algorithm. Since RANSAC could excessively reduce the data set, we try to ensure a sufficient number of matches (eight – double the minimum number of points needed to evaluate any homography) by incrementing the acceptance error threshold in RANSAC until the criterion is met or the error threshold becomes too large. Finally, the parameters of the mapping homography H are evaluated from the robust point matching set. Because registration can fail in featureless areas, we check that the homography is reasonable (i.e., not degenerate or scaled by very small or large values). For videos where registration fails (e.g., due to insufficient matches or degenerate homography), we fall back to using the position given by the marker tracking.

Transmission

The transmission module is responsible to transmit marker-based camera poses and video streams, from the sender to the receiver. This information is not necessary streamed together, and a packet can contain camera pose only, video only, or a combination of the two. Transmission is performed over UDP. In the current implementation, video is read at 480×360 resolution, using JPEG encoding for each frame. Each video packet, sent at a rate of 10 Hz using a shared wireless 802.11g network, is typically 5–30KB, and thus within the capacity of a single UDP packet. On the receiving side, the system receives a number of input video sequences and corresponding estimates of the camera pose relative to the panorama. This information is then used by the receiver to correctly display the various video streams within the static panorama.

Display

The renderer integrates multiple videos from multiple cameras, displaying them in a 3D scene with the panoramic image as background (Figures 1 and 5). As the renderer operates on the information received from the sender, the rendering varies depending on the type of packet received and is computed for each camera separately.

(a) The real environment. **(b) The virtual environment.**

Figure 6. Real environment and virtual copy used for the experiment.

If the received packet contains the marker-based estimate of the camera pose and a video frame, then the renderer displays the video inset using a projective texture based on the camera position returned by the marker tracking. The texture is projected on the six faces of the cube-map, and it is applied to a camera volume which is shaped by the intrinsic parameters of the smartphone's front camera (Figure 5(a)). If the receiver receives only a video frame, then the feature-based camera tracking needs to be performed to estimate the camera position. When this is done, the renderer applies the incoming video as texture of an extended plane that coincides with the face of the cube-map that is selected by the SIFT matching process. The estimated homography is converted into a texture coordinate matrix, and this plane is rendered with the video textured on it over the original texture from the static panorama. To obtain visually pleasant video overlay, the incoming video texture is blended into the panorama using alpha blending around the borders of the video texture. Furthermore, as the color balance of the smartphone's front camera might be noticeably different from the camera used to captured the panorama images, we ensured the white balance was the same by computing beforehand an overall static color balance correction using example images (Figure 5(b)).

USER STUDY

Experimental Design

Our user study aimed to assess the extent to which viewers are able to perceive and act on varying video modes over two spatial visualization tasks. We compare our system with webcam and panoramic video, which, theoretically, display less and more spatial information, respectively. To be consistent with the webcam condition that features the usual single camera, we test our system with only a single smartphone. For both webcam and PanoInserts conditions, we used the iPhone 4 front-facing camera in portrait mode to capture and transmit video. While our system is able to support several smartphones running in parallel to populate a static panorama with dynamic inserts, it is critical to assess the quality of our fundamental approach without being diverted into assessing how this may change as the number of dynamic inserts increases. We reserve this for future work. We used a PointGrey Research Ladybug3 camera for the panoramic condition.

In both tasks, the participants viewed a remote meeting room featuring a "horseshoe-shaped" table arrangement surrounding a central table on which the appropriate camera could be positioned (Figure 6(a)). We used stands to ensure that video from the Ladybug3 or iPhone camera was acquired from the

(a) Panoramic. (b) PanoInserts. (c) Webcam.

Figure 7. Representations of the remote room using each system. Panoramic and PanoInserts videos are cropped for illustration purposes.

same position. All the cameras were initially facing the center of the room. Both tasks involved object placement: either placing virtual objects to match the locations of real objects as perceived from the video stimuli, or the reverse of this, which is instructing a confederate to place real objects as seen through video stimuli to match the locations of virtual objects. The set of objects consisted of typical things one may find in an office or at home, and varied in size from $10cm^3$–$50cm^3$, and in color and shape.

The first task required participants to view a remote meeting room in which thirteen objects were positioned on tables around the room. Participants were required to determine where these objects were positioned in the room, and to use an interactive virtual model of the room to position the objects' virtual counterparts accordingly. A scaled virtual model of the room was created using Autodesk 3DS Max, which was then loaded into the experimental interface developed using Unity . At the beginning of the experiment, the virtual objects were located at the center of the virtual model shown in Figure 6(b). The virtual objects could be repositioned by dragging-and-dropping using the mouse. As the angular separation between the leftmost and rightmost objects was approximately 180°, participants in both the webcam and PanoInserts modes required the 30°camera to be rotated during the task to reveal different areas of the room. Hence, in these two conditions, participants could instruct a confederate located at the remote meeting room to rotate the camera.

The second task reversed the real-to-virtual object placement done in the first task, and required participants to match the positions of real objects in the meeting room with those presented in the same virtual model as used in the first task. Participants viewed a non-interactive virtual model of the remote meeting room in which the same thirteen objects were positioned (differently to how they were positioned at the meeting room in the first task) as shown in Figure 6(b). Participants instructed a confederate at the meeting room to place objects to match the virtual layout. To minimize the influence of the confederate's behavior, they could only follow direct instruction from the participant such as, "place the object half-way along the table directly behind you", and could not help in any other way. The confederate strictly and literally followed such directions given by the participant with minimal verbal interaction. As in the first task, participants could also ask the confederate to rotate the camera in the webcam and PanoInserts modes in order to reveal different parts of the scene.

Data Collection

These two tasks intended to explore the accuracy with which participants can correctly obtain a spatial understanding of a remote environment over the three modes. In both tasks, we measured object placement error, task completion time and, in the webcam and PanoInserts conditions, requested camera movements. After the participant had finished each task, we measured the positional (2D horizontal) error of either the virtual objects as placed by the participant in the virtual room (first task), or the real objects as placed by the confederate as per the participant's instructions in the real room (second task). Following the experiment, participants completed the standard System Usability Scale (SUS) questionnaire, which gathered subjective assessments of usability of the three systems, for the full set of questions, please refer to Brooke [5].

Hypotheses

For both tasks, we expected task performance to vary according to the spatial information each mode theoretically preserves. Hence, we expected participants using the panoramic video mode to be able to both place objects (virtual object placement task) and instruct objects to be placed (real object placement task) more accurately than participants using PanoInserts. In turn, we expected participants using PanoInserts to be more accurate than those in the webcam condition. Regarding number of camera movements, we expected the participants using PanoInserts to require fewer than those in the webcam condition due to the presence of the static panorama background. Note that the panoramic condition requires zero camera moves as the whole panorama is dynamic. Regarding task completion time, we expected participants using panoramic video mode would require the least time than those in the other two conditions. Our expectancy of the usability scores as measured by the SUS questionnaire were less clear, as the panoramic representations of space as presented by both PanoInserts and the panoramic systems may be unfamiliar to participants and take some acclimatization that may influence the scores. We did expect, however, that all three video modes would be ranked reasonably highly in terms of overall usability.

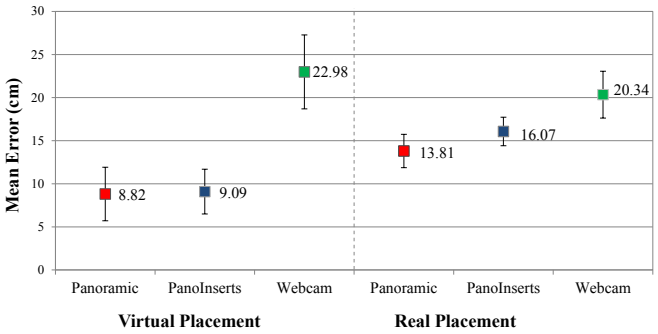


Figure 8. Mean object placement error and standard deviation for the three systems in both tasks.

Procedure

Participants performed both experimental tasks in a single video mode, so the experiment featured a between-subjects design in terms of the independent condition of video mode, and a within-subjects design in terms of task. A total of 36 unpaid participants took part (12 in each video mode condition), and we alternated the order in which the two tasks were performed to minimize the influence of learning effects. Participants were recruited from the staff and student population at our university.

Some participants had previously been in the meeting room used in our experiment. So, to ensure all participants had similar prior knowledge of the remote environment, we gave each as much time as they liked in order to walk around the room and become acquainted with the space. The participant was then brought into the lab where he/she was presented with two workstations: one displaying the video-mediated representation of the room in one of the three video modes (Figure 7), and the other displaying the virtual representation of the room. Objects were arranged in both real and virtual environments to the appropriate starting arrangement depending on which task was to be performed first. The participant was briefed on the appropriate task and on how he/she may instruct the confederate to move the camera in the webcam and PanoInserts condition and also to pick up and place objects if they were performing the real object placement task. Following completion of the task, the object placement errors along with time taken and number of camera moves (in webcam and PanoInserts conditions) were recorded. The room was then rearranged for the remaining task. The participant was briefed on the remaining task which they would then carry out, and data recording was subsequently performed. Finally the participant completed the SUS questionnaire.

RESULTS

Placement Accuracy

Figure 8 shows the mean error and standard deviation of object placement error for both tasks. We first address the task in which participants were required to place objects in the virtual environment to match the real environment's arrangement while viewing the meeting room using one of the three video modes. We calculated an Analysis of Variance (ANOVA) using SPSS with the two factors of video mode and object and the dependent variable of placement error. A significant main effect of video mode was found ($F_{(2,11)} = 66.555$, $p < 0.001$), with a lower error for the panoramic (*Mean*, $M = 8.82\ cm$) and PanoInserts ($M = 9.09\ cm$) conditions than for the webcam condition ($M = 22.98\ cm$). Post-hoc Tukey tests revealed non-significant differences between the panoramic and PanoInserts conditions ($p = 0.979$), and significant differences between the webcam and panoramic conditions ($p < 0.001$). A main effect was found between PanoInserts and webcam conditions ($p < 0.001$). A significant main effect of object was found ($F_{(2,11)} = 3.015$, $p < 0.001$).

We now focus on the task in which participants were required to instruct a confederate to place objects in the real environment to match the virtual environment's arrangement while viewing the meeting room using one of the three video modes. Similarly, we calculated an Analysis of Variance (ANOVA) using SPSS with the two factors of video mode and object and the dependent variable of placement error. As above, a significant main effect of video mode was found ($F_{(2,11)} = 4.849$, $p = 0.008$), with a lower error for the panoramic ($M = 13.81\ cm$) and PanoInserts ($M = 16.07\ cm$) conditions than for the conventional webcam condition ($M = 20.34\ cm$). Post-hoc Tukey tests again revealed non-significant differences between the panoramic and PanoInserts conditions ($p = 0.555$), and significant differences between the webcam and panoramic conditions ($p = 0.007$). However, no main effect was found between PanoInserts and webcam conditions ($p = 0.112$). The main effect of object was also significant ($F_{(2,11)} = 3.022$, $p = 0.001$).

Table 1. Mean time to complete (sec) and required camera moves for the three systems in both the virtual object placement (VOP) and real object placement (ROP) task.

	Time to Complete		Camera Moves	
	VOP	ROP	VOP	ROP
Panoramic	198.37	444.55	N/A	N/A
PanoInserts	395.19	538.32	7.58	7.41
Webcam	169.92	561.01	8.5	8

Time to Complete

Table 1 reports the mean time to complete each task in each video mode. We first analyze the virtual object placement task. We computed an Analysis of Variance (ANOVA) using SPSS with the single factor of video mode and the dependent variable of total time to complete the task. Video mode was not found to be a significant factor ($F_{(2,1)} = 1.356$, $p = 0.272$). We now address the real object placement task. Similarly, we calculated an Analysis of Variance (ANOVA) using SPSS with the single factor of video mode and the dependent variable of total time to complete the task. As above, no main effect was found ($F_{(2,1)} = 1.794$, $p = 0.190$). We note that there is a large variance between participants, and we briefed participants to complete the tasks with object placement accuracy in mind as opposed to speed.

Required Camera Moves

For the PanoInserts and webcam conditions we also collected the total number of camera moves required by each participant while completing the two tasks. Table 1 reports the mean number of camera moves for each mode. Regarding the virtual object placement task, we calculated an Analysis of Variance (ANOVA) using SPSS with the single factor of video mode and the dependent variable of number of camera moves requested by the participant to complete the task. No main effect was found ($F_{(1,1)} = 0.957$, $p = 0.339$). Focusing on the real object placement task, an ANOVA also did not uncover a significant different between conditions ($F_{(1,1)} = 0.542$, $p = 0.470$).

Finally, for both webcam and PanoInserts conditions we computed the correlation coefficient r between the participants' requested camera moves and the participants' mean error. A moderate negative correlation was found for PanoInserts in both the virtual object placement task ($r = -0.664$) and the real object placement task ($r = -0.324$). However, for the webcam condition the correlation coefficient reveals a weak positive correlation for both the virtual object placement task ($r = 0.126$) and the real object placement task ($r = -0.104$). Implications of these results are discussed in the next section.

SUS Questionnaire

Following the experiment, each participant completed the standard System Usability Scale (SUS) questionnaire. All modes obtained positive results, with the webcam condition obtaining the best score ($SUS = 82.5$), followed by the panoramic ($SUS = 77.29$) and PanoInserts ($SUS = 73.54$) conditions. Based on these results, and following the analysis technique suggested in [14], the webcam system can be classified as Rank A system (out of six possible letter-grade ranks varying from A to F), while both PanoInserts and the panoramic mode can be classified as Rank B systems.

DISCUSSION

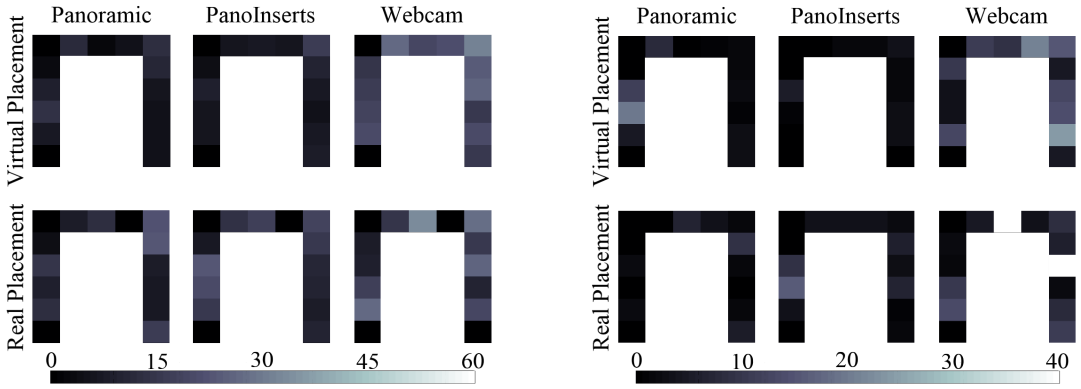
Task Performance

The results from our user study reveal information into the way participants were able to spatially perceive and act on information presented in the varying video modes. In both tasks, panoramic video and PanoInserts enabled greater accuracy than webcam video when positioning objects. This finding is in accord with each video mode's relative degree of spatiality as hypothesized, and suggests that both fully- and partially-dynamic panoramic representations of space can encode information that people can intuitively understand and act upon.

Exploring the number of camera moves participants performed reveals information about how participants went about completing the tasks. As the panoramic condition did not require camera movement, here we discuss only the webcam and PanoInserts conditions. While not found statistically significant in our analysis, participants in the PanoInserts condition performed fewer camera movements than those in the webcam condition (Table 1). A moderate negative correlation between camera moves and mean error was also noted for PanoInserts, but not for the webcam mode. This indicates that PanoInserts users were able to incrementally decrease placement error through camera repositioning. The same does not apply to the 2D video case, as its correlation coefficients reveal a weak positive correlation for the virtual object placement task. This suggests that participants could apply the additional spatial information presented in PanoInserts to improve their spatial reasoning ability of the remote location. Concerning the time to complete the tasks, PanoInserts' users systematically required more time to ultimate their tasks. This can be justified by the fact that the system performances was influenced by switching the camera tracking mode, which we will refine in future versions of the system.

Placement accuracy differed in between the two tasks, with the virtual object placement task resulting in a relatively smaller error and standard deviation than the real object placement task. While the two tasks were complementary and both relied on spatial reasoning, they differed in some key aspects. When positioning virtual objects to match those viewed in the physical space, participants observed a visual representation of the real objects spread over the tables in the room from a perspective similar to being in the room. This embedded additional spatial cues in the video stimuli, provided by the objects' relative locations and the camera's viewpoint. This resulted in some participants instructing the confederate to move the camera "in between" certain objects, effectively restricting placement error to greater extent than in the real object placement task. Contrastingly, in the task requiring positioning of real objects to match those in the virtual space, participants were presented with a top-down virtual reference representation from which to work from that was more similar to the perspective of a CCTV camera than it is to being in the room. So, participants could use only environmental cues to estimate where an object should be placed. They could also use objects that they had just placed, but error could accumulate. This allowed more room for incorrect placement.

Hence, the two tasks presented qualitatively different reference stimuli from which the task of positioning objects is then required to be carried out. The accuracy results shown in Figure 8 show that participants found the real object placement task more difficult than the virtual object placement. Exploring the impact of task further, we calculated three post-analysis single-factor ANOVAs using task as factor, and data from a single video mode. Significant differences were found between tasks in panoramic ($p = 0.028$) and PanoInserts ($p = 0.001$) conditions, but not in the webcam condition ($p = 0.607$), where the real object placement task actually attained slightly greater accuracy. We note, then, that participants found the conversion between a person-perspective view to a top-down representation (as in the virtual object placement task) easier than they found the reverse. However this depends on the spatial richness of the stimuli, and does not hold if the spatial nature of the perspective view is impoverished as in the webcam condition. We now further explore the differing spatial representations offered by the three video modes.



(a) Mean error and room location.

(b) Mean error variance and room location.

Figure 9. How mean error and error variance varies over the room. Each tile represents a portion of the desk.

Spatial Representation

When displayed on a standard flat display, panoramas represent a surrounding environment in a way that is often not intuitively clear, and differs considerably from how we visually perceive space in normal life. Panoramas present space at a greater field-of-view than the human visual system does, so the viewer has to cognitively translate that representation before understanding it. On the contrary, conventional webcam video presents space with a field-of-view that is less than human vision, so is directly intuitive for the viewer. While our experimental results show that people can understand the panoramic content and use it to complete the tasks efficiently, there are likely to be better ways of presenting it. In the future, we intend to explore both hardware and software approaches to this problem. Displays such as Global Imagination's spherical Magic Planet or immersive projection technologies such as the CAVE™ or head-mounted displays are able to complement the acquisition technology and present panoramic content in a way that preserves its surrounding nature. Software approaches to enable clearer representation of the spatial mapping between panorama and environment may be achieved through visually-correcting interesting portions of the panorama through perhaps a "pop-out" metaphor, or by presenting the entire panorama in a virtual environment as seen in [19].

As stated previously, participants visited the experimental meeting room prior to the experiment, and were also presented with the virtual model during experiment, helping them to form an idea of the spatial layout of the room. During the experiment, participants were required to translate between a top-down virtual model of the room and a first-person perspective video representation of the room. These two visualizations present space differently. Specifically, the distortion present in the video modes varies across the image, so that the screen-space distance between two pixels in the video that map to two points in the physical room may not be equal to the distance of another two other points in the room of equal physical distance. This depends on the distance of the objects to the camera, and is due to camera foreshortening, which usually results in more error around the corners of a camera view.

We assessed the influence of object position post-hoc, and present Figure 9. The plots visualize the horseshoe-shaped table in the experimental room, and encode mean object placement error and error variance as a heat-map. Both error and error variance is seen to vary across the environment, with the greatest readings localized around upper-right corner and left side of the tables. The varying visual distortion inherent in video is likely to influence object placement accuracy around the 180° range. The error variance across objects (Figure 9(a)) is noticeably larger for the webcam condition than the other two conditions, suggesting that participants using it were performing the spatial reasoning task based on poorer information and were less accurate as a result.

Usability

All the three systems obtained a high SUS scores, with participants rating the webcam mode highest ($SUS = 82.5$, Rank A), followed by panoramic ($SUS = 77.29$, Rank B) and PanoInserts ($SUS = 73.54$, Rank B) modes. The webcam system's higher score is likely due to its familiarity with participants. Also regarding usability, it was interesting to observe how participants went about the tasks in each condition. Participants in the webcam condition often required an initial camera rotation from one corner to the room to the other, indicating that they were unsure as to where the camera was facing in the room. Also, several participants in the webcam condition became confused with regards to which direction they needed to rotate the camera in order to see a different part of the room, which may indicate difficulty in self-localization in the remote location. These observations are supported by some of the post-experimental comments recorded. The majority of participants that experienced PanoInserts considered the static panorama to be a valuable resource providing spatial information about camera heading and object location.

CONCLUSIONS AND FUTURE WORK

We have presented PanoInserts, a system allowing users to rapidly assemble a set of cameras to generate a panorama with live inserts for use in teleconferencing applications. We conducted a user study assessing how our system is able to support collaboration in spatial reasoning tasks, comparing performance with traditional webcams and expensive panoramic

cameras. Results indicate that our system performs comparably with fully-panoramic video, and better than webcam video conferencing in tasks that require a surrounding representation of the remote space. This suggests that our approach lies between fully-panoramic and webcam-based video both in terms of its technical characteristics and device accessibility, and also in terms of the richness of the conveyed spatial information that users can demonstrably understand and act upon. We have discussed issues relating to the problematic visual perception of panoramas due to varying distortion according to depth, and we intend to investigate methods for displaying panoramic content in a visually-intuitive manner. We will also extend the system to support bidirectional communication and groups at more than two locations. We will also extend the system to work in highly-dynamic environments such as outdoors, which will require enhanced camera tracking and video stabilisation as well as directly using panoramic cube-maps retrieved from online map data. Finally, we will investigate further modes of panoramic telecommunication in scenarios featuring more complex and unpredictable social interaction.

ACKNOWLEDGEMENTS

This work is supported by the BEAMING project (**www.beaming-eu.org**) which is funded by the European Commission under the FP7 ICT Work Programme.

REFERENCES

1. Agarwala, A., Zheng, K. C., Pal, C., Agrawala, M., Cohen, M., Curless, B., Salesin, D., and Szeliski, R. Panoramic video textures. *ACM Transactions on Graphics 24*, 3 (July 2005), 821–827.
2. Benford, S., Brown, C., Reynard, G., and Greenhalgh, C. Shared spaces: transportation, artificiality, and spatiality. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, ACM (1996), 77–86.
3. Benford, S., Greenhalgh, C., Reynard, G., Brown, C., and Koleva, B. Understanding and constructing shared spaces with mixed-reality boundaries. *ACM Transaction on Computer-Human Interaction 5*, 3 (Sept. 1998), 185–223.
4. Billinghurst, M., Poupyrev, I., Kato, H., and May, R. Mixing realities in shared space: An augmented reality interface for collaborative computing. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, vol. 3, IEEE (2000), 1641–1644.
5. Brooke, J. SUS: A quick and dirty usability scale. In *Usability evaluation in industry*, P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, Eds. Taylor and Francis, London, 1996.
6. Brown, M., and Lowe, D. G. Recognising panoramas. In *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, IEEE Computer Society (2003), 1218–1226.
7. Chen, M. Leveraging the asymmetric sensitivity of eye contact for videoconference. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, ACM (2002), 49–56.
8. Cutler, R., Rui, Y., Gupta, A., Cadiz, J., Tashev, I., He, L.-w., Colburn, A., Zhang, Z., Liu, Z., and Silverberg, S. Distributed meetings: a meeting capture and broadcasting system. In *Proceedings of the ACM International Conference on Multimedia* (2002), 503–512.
9. Diverdi, S., Withert, J., and Hllerert, T. Envisor: Online environment map construction for mixed reality. In *Proceedings of IEEE VR Conference* (2008).
10. Fiala, M., Green, D., and Roth, G. A panoramic video and acoustic beamforming sensor for videoconferencing. In *Haptic, Audio and Visual Environments and Their Applications* (October 2004), 47–52.
11. Fischler, M. A., and Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM 24*, 6 (1981), 381–395.
12. Hauber, J., Regenbrecht, H., Billinghurst, M., and Cockburn, A. Spatiality in videoconferencing: trade-offs between efficiency and social presence. In *Conference on Computer Supported Co-operative work*, ACM (2006), 413–422.
13. Kauff, P., and Schreer, O. An immersive 3d video-conferencing system using shared virtual team user environments. In *Proceedings of the 4th international conference on Collaborative virtual environments*, ACM (2002), 105–112.
14. Lewis, J., and Sauro, J. The factor structure of the system usability scale. *Human Centered Design* (2009), 94–103.
15. Li, Y., Wang, Y., Huang, W., and Zhang, Z. Automatic image stitching using SIFT. *International Conference on Audio Language and Image Processing* (2008), 568–571.
16. Lowe, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Visiono 60*, 2 (2004), 91–110.
17. Majumder, A., Seales, W. B., Gopi, M., and Fuchs, H. Immersive teleconferencing: a new algorithm to generate seamless panoramic video imagery. In *Proceedings of the International Conference on Multimedia*, ACM (1999), 169–178.
18. Mikolajczyk, K., and Schmid, C. Scale and affine invariant interest point detectors. *International Journal of Computer Vision 60*, 1 (2004), 63–86.
19. Mulloni, A., Seichter, H., Dünser, A., Baudisch, P., and Schmalstieg, D. 360 degrees: panoramic overviews for location-based services. In *Proceedings of the Annual Conference on Human Factors in Computing Systems*, ACM (2012), 2565–2568.
20. Rousseeuw, P. J. Least median of squares regression. *Journal of the American Statistics Association 79*, 388 (1984), 871–880.
21. Rui, Y., Gupta, A., and Cadiz, J. Viewing meeting captured by an omni-directional camera. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2001), 450–457.
22. Schaffalitzky, F., and Zisserman, A. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proceedings of the European Conference on Computer Vision*, Springer-Verlag (2002), 414–431.
23. Steedly, D., Pal, C., and Szeliski, R. Efficiently registering video into panoramic mosaics. In *Proceedings of the International Conference on Computer Vision*, vol. 2 (2005), 1300–1307.
24. Steptoe, W., Normand, J.-M., Oyekoya, O., Pece, F., Giannopoulos, E., Tecchia, F., Steed, A., and Slater, M. Acting rehearsal in collaborative multimodal mixed reality environments. *Presence: Teleoperators & Virtual Environments 21*, 4 (2012), 406–422.
25. Suen, S. T., Lam, E. Y., and Wong, K. K. Photographic stitching with optimized object and color matching based on image derivatives. *Optics Express 15*, 12 (2007), 7689–7696.
26. Szeliski, R. Image mosaicing for tele-reality applications. In *Proceedings of the IEEE Workshop on Applications of Computer Vision* (1994), 44–53.
27. Szeliski, R. Image alignment and stitching: a tutorial. *Foundations and Trends in Computer Graphics and Computer Vision 2* (2006), 1–104.
28. Szeliski, R., and Shum, H.-Y. Creating full view panoramic image mosaics and environment maps. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co. (1997), 251–258.
29. Vertegaal, R., Weevers, I., Sohn, C., and Cheung, C. GAZE-2: conveying eye contact in group video conferencing using eye-controlled camera direction. *Proceedings of the conference on Human factors in computing systems* (2003), 521–528.
30. Wagner, D., Mulloni, A., Langlotz, T., and Schmalstieg, D. Real-time panoramic mapping and tracking on mobile phones. In *Proceedings of the IEEE VR Conference*, IEEE Computer Society (2010), 211–218.
31. Wu, C. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). **http://cs.unc.edu/~ccwu/siftgpu**, 2007.